Automated Semantics and Topology Representation of Residential-Building Space Using Floor-Plan Raster Maps

Bisheng Yang, Tengping Jiang, Weitong Wu, Yuzhou Zhou, and Lei Dai

***Abstract*—Automatically representing the semantics and topology of indoor building spaces from floor-plans is necessary for many applications, such as architectural design and indoor renovations. Extensive studies have investigated reconstructing indoor spaces with semantics and topology using professional means (e.g., laser scanning and photogrammetry). Floor-plan raster maps are widely and freely available for various purposes. Nevertheless, there is little research on the semantic and topological representation of indoor elements from floor-plan raster maps. To fill this gap, we propose a method of automatically representing the semantics and topology of indoor spaces from floor-plan raster maps. The proposed method first identifies basic geometric primitives from floor-plans using a learning-based hierarchical segmentation approach. Second, the relationship between the detected geometric primitives is assembled into the planar structure representation with topological data using mixed integer programming. Finally, the floor-plan graph structure is checked and optimized to maintain consistency with a polygonal coordinate descent strategy, resulting in a correct representation of the semantics and topology of the indoor space. Comprehensive evaluations demonstrate that the proposed method effectively achieves superior performance in three different datasets. The proposed method allows for 3D model popups for better visualizations and direct architectural model manipulations of the interior building layouts computation.**

***Index Terms*—Deep learning, floor plan, indoor space, semantics and topology representation.**

I. Introduction

RECONSTRUCTING indoor building spaces with semantics and topology has received ample attention in the fields of laser scanning, photogrammetry, and computer graphics. Such reconstructions pertain to a wide spectrum of applications, such as indoor navigation, emergency evacuation, and augmented reality [1]. Automated structured representations of indoor residential-building scenes are a popular topic [2]. Many studies have sought to represent the semantics and topology of indoor residential-building spaces by professional means, such as optical images [3], [4], [5] and 3D scanning data [6], [7], [8], [9], [10].

However, because of the unique conditions of indoor spaces, such as varied light conditions, pedestrian disturbances, and occlusions [11], optical imagery-based methods suffer from poor image quality (e.g., distortion). As such, they are limited in their ability to represent the semantics and topology of indoor elements. Compared with imaging-based systems, 3D scanning techniques are more robust to illumination variations and data distortions [12], with more accuracy and robustness to geometrical reasoning. However, 3D scanning data is not widely available and it requires advanced knowledge to deal with the huge volume of data [13].

In recent decades, residential-building floor-plan raster maps have become widely and freely available on the internet. For example, many real estate companies provide building floor-plans on the internet to sell houses [14]. Unfortunately, residential-building floor-plan raster maps are mainly used for visualization purposes and thus lack the semantics and topology of indoor elements. To automate the construction of the topology and semantics of residential floor-plans, deep learning techniques have been proposed, offering revolutionary improvements to the detection of low-level information. However, the task of holistic high-level geometric structural reasoning for floor-plan raster maps remains challenging and is usually only possible in the hands of professionals. Hence, rapid and robust methods are urgently required to represent indoor building spaces with topology and semantics from widely available residential-building floor-plan raster maps.

The structured representation of the indoor architecture from a floor-plan image is even more challenging given the irregular layout, uneven element thickness, and non-Manhattan walls in indoor environments. To overcome these challenges, we propose a novel three-stage approach that integrates modified convolutional neural networks (CNNs), mixed integer programming (MIP), and an optimization strategy to reconstruct the semantics and topology from floor-plan raster maps. The main contributions of the proposed method are threefold. First, geometric primitives are detected using deep learning-based hierarchical segmentation, which takes raw floor-plan images

Manuscript received 7 December 2021; revised 3 March 2022 and 31 May 2022; accepted 26 August 2022. Date of publication 15 September 2022; date of current version 20 September 2022. This work was supported in part by the National Science Fund for Distinguished Young Scholars under Grant 41725005, in part by the Key Project of National Natural Science Foundation of China under Grant 42130105, and in part by the National Natural Science Foundation of China under Grant 41071268. *(Corresponding authors: Bisheng Yang; Tengping Jiang.)*

The authors are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China, and also with the Engineering Research Center for Spatio-Temporal Data Smart Acquisition and Application, Ministry of Education, Wuhan 430079, China (e-mail: bshyang@whu.edu.cn; jiangtp_3d@whu.edu.cn; weitongwu@whu.edu.cn; zhouyuzhou@whu.edu.cn; dailei@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3205746

as input data and recognizes low- to mid-level semantic information to identify three types of geometric primitives (viz., junction corners, boundary edges, and room regions). Second, a constrained MIP approach is developed to fuse identified geometric primitives and their relationship information into a planar graph. In this step, geometric errors, primitive-wise relationships, and prior information are taken into account in an extended objective function to ensure the initial indoor architecture vectorization. Third, the floor-plan structured graph inference problem is solved using a polygonal coordinate descent strategy, which is capable of effectively repairing the topological inconsistencies of the resulting polygon. The performance of the proposed method was evaluated with two publicly accessible benchmarks—R2V [15] and R3D [16]—and a dataset collected from Beike (www.ke.com). The results show that the proposed method performs well at representing the semantics and topology of indoor spaces, and that it is efficient in terms of time and costs.

II. Related Work

This section presents a brief review of recent progress regarding two major tasks: floor-plan image parsing and structuring.

A. Floor-Plan Image Parsing

While floor-plan image parsing is generally straightforward for humans, automatically recognizing semantics is challenging due to the high diversity of floor-plans. In the past, manually interpreting multiple floor-plan raster maps were labor-intensive, time-consuming, and costly, due to their high diversity. Fortunately, rule-based heuristic approaches provide a solution to automatically parse floor-plan raster maps by utilizing image processing (e.g., morphological filtering [17], graphic recognition [18], or Hough transformation [19]). However, floor-plans were created by different people for different purposes, this means that the drawing conventions can be extremely flexible. Rule-based methods struggle to handle floor-plans with complex annotations and high diversity.

Most methods follow a traditional pipeline that starts with the separation of textual data (e.g., the dimensions and room labels) from graphical data (e.g., line primitives) [20]. The floor-plan elements are then identified by relying on hand-crafted rules. Clearly, the level of performance is error-prone due to a lack of generality when handling diverse conditions. Machine learning algorithms [21], [22] have emerged, which can avoid the limitation that existing style-dependent heuristics are *ad hoc*. Indoor elements are detected by feature descriptors and off-the-shelf classifiers. The output is then converted into vector data [23]. Based on the combinatorial maps and their duals, Yang and Worboys [24] enhanced the compact representation of geometric and semantic information. Approaches using machine learning algorithms have shown great potential for floor-plan analysis and take expressive generality among various styles. However, each approach is limited to only a few classes, and results in largely abstracted primitives [25].

With the success of deep learning [e.g., CNN and graph convolutional networks (GCNs)], researchers have recently begun to explore powerful deep learning tools for floor-plan image analysis. In [26], a fully convolutional network (FCN) was trained to identify the walls in a given floor-plan image. A faster R-CNN framework was then adopted to locate the other symbols. Similarly, Yamasaki et al. [27] quickly retrieved ideal apartments of similar structures based on semantic pixels, which were classified by an FCN. Then, the pixels associated with a semantic class label were taken to form a graph model. To maximize the number of obtained structural elements, Zeng et al. [28] constructed a hierarchy of floor-plan elements and designed a multitask network with two tasks: one to learn to predict boundary elements, and the other to predict room types. After preprocessing, Renton et al. [29] converted floor-plan images into a graph and applied a GCN to classify the symbols and objects. However, the final output of the approach was blurry, as they performed pixel-level segmentation, which is unsuitable for practical applications (i.e., modifying the structure of indoor space and redesigning flexibly) due to lack of vectorization information.

B. Floor-Plan Image Structuring

While segmentation from a floor-plan image has been a popular problem, which represents individual primitives as a set of pixels with corresponding labels [30], structural inferences from building-plan raster maps are particularly meaningful and an effective solution for models of indoor buildings.

With a known fixed topology, vectorized reconstruction from indoor point clouds is one of the most successful examples of graph structure reconstruction [31]. However, point clouds generated by various means usually contain more clutter and missing regions, which significantly enhance the difficulty of indoor vectorized reconstruction [32]. Classical architecture vectorization from a single RGB image is the closest to our work, and involves utilizing architectural shape grammars or learning structural regularities from examples [33], [34]. In contrast, the topology of floor-plan images is unknown and varies in each case. Nevertheless, establishing the semantic relationships and topology with the above-mentioned two tasks was a source of inspiration for our work.

As for the semantics and topology representation of indoor elements, few usable solutions exist. To the best of the authors' knowledge, the work in [15] was one of the first to obtain topological and geometric representations of floor-plan raster maps. They trained an available CNN to detect the junctions and applied integer programming (IP) to obtain vector data with the Manhattan assumption. FloorNet [35], which consists of three deep neural network architectures, was introduced to turn an RGBD video into pixel-wise floorplan geometry and semantics information. Subsequently, an existing IP formulation was adopted to recover vector-graphics representation. Similarly, the method in [36] is also divided into recognition and reconstruction, the performance of floor-plan segmentation and vectorization is improved by introducing multimodal information (e.g., text, symbols, and scale of floor-plans). Instead of corner detection, the work in [37] recovered critical structure details by relying on a space-partition strategy and energy

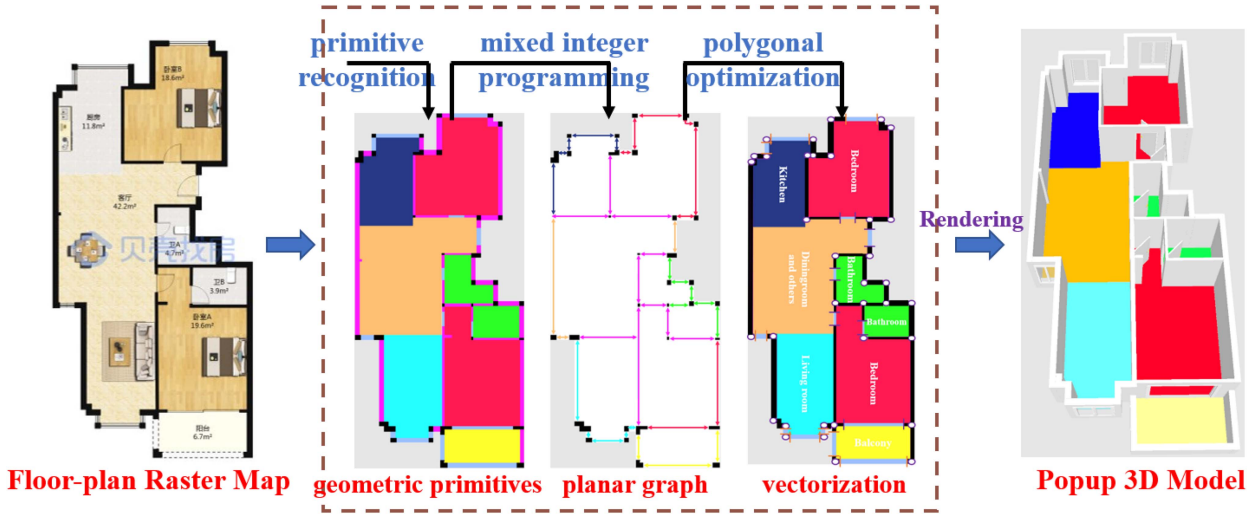

Fig. 1. Workflow of the proposed method for indoor building space semantics and topology representation.

minimization. Wu et al. [38] proposed a data-driven technique for floor-plan analysis that learns the design rules from the original dataset and achieves more attractive plausibility than the above-reported works. Wu et al. [39] adopted a two-stage pipeline that imitates the professional design process for automatically and efficiently computing residential-building layout information with certain boundaries. Similarly, Wu et al. [40] vectorized building elements and repaired the topological inconsistencies of the vectorization outputs for mapping and modeling indoor scenes. However, the above-reported methods consider only 2D geometry and ignore international standards and usability. Jang et al. [41] converted vector results obtained from floor-plans into CityGML and IndoorGML.

More recently, GCNs with numerous variations have been applied to graph data structures, showing excellent performance. A floor-plan can also be converted into a graph by treating cell regions as nodes and constructing an adjacency matrix based on the adjacency among the regions [25]. Various graph-related methodologies have been applied to floor-plan analysis. Hu et al. [42] proposed a novel yet simple framework, Graph2Plan, for indoor layout graph representation that focuses on graph inference problems in the context of reconstruction. Nauata and Furukawa [33] encoded the constraint into the graph structure of its relational networks for axis-aligned bounding boxes of room productions. Later, based on Conv-MPN [34], [43] proposed a layout refinement network that integrates a graph-constrained relational generative adversarial networks (GAN) and a conditional GAN. To obtain realistic 3D interior models, the Plan2Scene [44] proposed an efficient GCN architecture that inferred textures for unobserved surfaces based on residential floor-plans and a set of associated photos. To realize indoor localization services that require the indoor spatial information and the relationships between indoor spaces, Yang et al. [45] proposed a semantics-guided method for deducing topology of indoor spaces. In their developed hierarchical framework, a GCN-based method was used to model the long-range relationships among primitives in the real world.

Although the reported methods based on floor-plans have achieved satisfactory results with simple indoor building layouts, automatic and correct representations of the semantics and topology of indoor building spaces from building floor-plan raster maps that do not follow the Manhattan geometry remain challenging. To accurately recover semantic and vectorized information from floor-plan raster maps, this article proposes a method that combines CNNs and MIP to meet this challenge. Specifically, CNNs extracted low- to mid-level semantic information, and MIP consolidated all the information and reconstructed a planar graph with topology representation.

III. Methodology

Fig. 1 illustrates the pipeline of the proposed method. The proposed method first identifies a set of simple geometric primitives with semantics. Then, an MIP assembles the information of the primitives into a planar graph. Finally, the vector information is further enriched and refined with global energy optimization to produce a floor-plan with the correct topology and semantics.

A. Identifying Geometric Primitives With a Hierarchical Segmentation

Identifying geometric primitive involves recognizing the layout semantics from floor-plan raster maps. Most identification methods output the pixel-level segmented masks with rugged boundaries and outline a coarse layout of the indoor space [40]. Instead of directly identifying the primitives, the proposed method solves the aforementioned problem by predicting scene cues. To parse floor-plan images for reliable primitive identification, semantic segmentation network is selected as the base network. For reliable floor-plan image parsing, a hierarchical segmentation that imitates human perception is proposed. Specifically, the floor-plan elements are organized in a hierarchy. As shown in Fig. 2, the hierarchical segmentation pipeline includes three steps: boundary and room elements classification, semantic information extraction, and geometric primitive

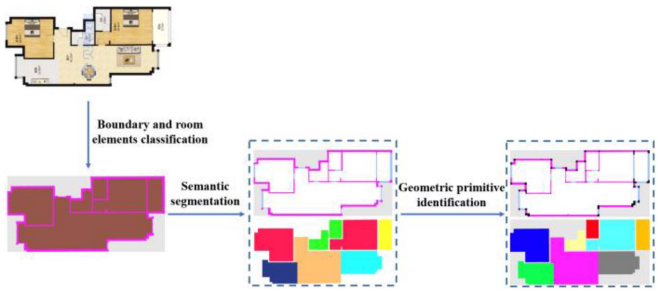

Fig. 2. Proposed hierarchical segmentation pipeline.

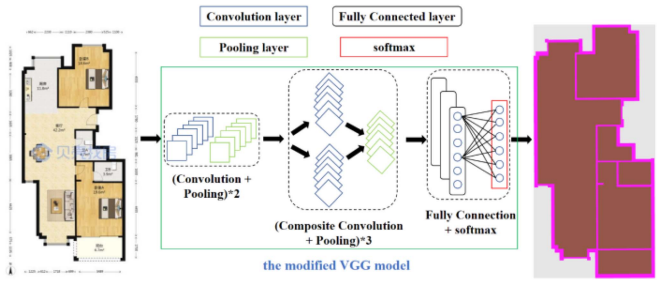

Fig. 3. Overview of boundary and room elements recognition. From left to right, an original input floor-plan raster map, the modified VGG network structure, and initial elements recognition output rendered by types.

identification. Each step takes the output of the previous step as input, three geometric primitive (junction corners, boundary edges, and room regions) are finally obtained.

1) Boundary and Room Elements Classification: As mentioned in previous studies [28], [40], floor-plan images typically depict boundary elements like walls and openings, as well as spatial elements like bedrooms, dining rooms, and living rooms. To handle rooms of different types and walls of nonuniform thickness, floor-plan images can be firstly segmented into boundary pixels or room pixels. Since pixels refer to either boundary elements or spatial elements, separating the boundary and room pixels can be considered as a binary classification problem. The boundary and room elements of each floor-plan raster map are processed using a modified segmentation network VGG model. The original VGG model has limited ability to extract complex and expressive features in images. However, it is difficult to adjust the parameters and the training time is quite long, due to the sequentially connected network structure. Therefore, a modified VGG model (see the green box in Fig. 3) is proposed by replacing many convolution layers with composite convolution ones [46].

In contrast, the modified VGG model reduces the number of convolutional layers and speeds up the extraction of image features. The input data undergo the combined action of the composite convolution layer, the pooling layer, and the fully connected layer to obtain a series of local features vectors, which are finally input into the classifier for converting the feature maps to an identification result. To achieve better performance, the intersection-over-union (IoU), instead of cross-entropy, is applied as a loss function of the modified model [47]. The IoU loss function is defined in (1). Fig. 3 illustrates an example of a floor-plan raster map, where the boundary (dotted in pink) and room (dotted in brown) elements are identified. The identification results are then further classified into the corresponding types of structural primitives.

$$\begin{cases} IoU = \frac{FPPixel_{id} \cap FPPixel_{gt}}{FPPixel_{id} \cup FPPixel_{gt}} \\ loss_{IoU} = -IoU \end{cases} \tag{1}$$

where $FPPixel_{id}$ represents the output pixels of the modified VGG model, and $FPPixel_{gt}$ denotes the ground truth pixels.

2) Semantic Information Extraction: In the second step, a parallel semantic segmentation network is introduced to process nonoverlapping but spatially-correlated boundary and room elements. As illustrated in Fig. 4, the segmentation network consists of four main components: a shared encoder, two parallel decoders, a spatial contextual module, and a joint prediction module. The two parallel decoders aim to predict boundary and room types. Given that CNNs are powerful tools for extracting image information, a shared encoder and two parallel decoders are adopted as the backbone network [28]. Following [48], the backbone is built by dropping the last deconvolution layer of ResNet [49] and appending three deconvolution layers in parallel for per-pixel semantic segmentation.

The proposed network takes a floor-plan image of size 512 $\times$ 512 as the input, and then encodes it into a 256 $\times$ 256 $\times$ 64-shaped matrix using the shared feature encoder. Next, the output of the feature encoder is input into the two separate decoders and processed by their subsequent components in parallel. The boundary pixels prediction branch decodes the shared features and fuses the features of the different layers into a feature matrix F_b (16 $\times$ 16 $\times$ 512). Similarly, the room pixel prediction branch outputs a semantic feature matrix F_t after the decoders. Finally, both of the features from the above-mentioned two branches are fetched and processed by the following module. They then output two feature matrices. One of the matrices, P_b (256 $\times$ 256 $\times$ 32), is used to detect the boundary elements. The other matrix, P_t (256 $\times$ 256 $\times$ 64), is a semantic feature matrix used to predict the labels for each room. To detect room areas and boundaries, a spatial contextual module is adopted to guide and bind the discovery of individual room candidates with a boundary-guided attention mechanism. Here, the module further leverages boundary contexts to obtain the attention weights for room prediction. Specifically, the boundary features are passed from the top decoder in the boundary prediction branch to another bottom decoder to produce features for integration with attention weights by making use of convolutional layers with four different direction-aware kernels [50]. The above-mentioned operation helps the decoder in the room-type prediction branch to maximize the learned contextual information and feature fusion for room-type predictions.

For multilabel tasks, it is important to balance the contributions of each task, because the number of pixels varies for different elements. Hence, a loss function is important to effectively balance the labels across and within branches. As the number of room-type pixels is far more than that of boundary pixels, a matured cross-and-within-task weighted loss [51] is introduced

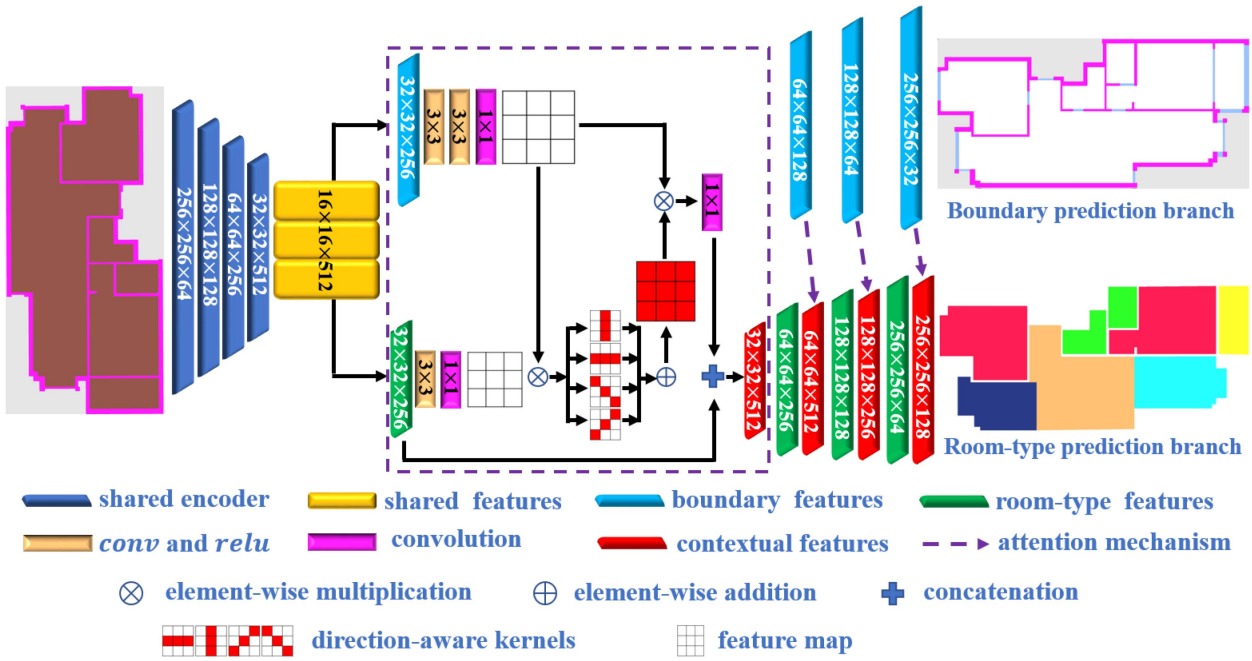

Fig. 4. Illustration of the proposed network for floorplan primitive extraction. The purple box is detail of boundary-guided attention mechanism (purple arrows).

to further balance the two prediction branches and the layout primitives within each branch. At the time of training, the overall weighted loss function $\mathcal{L}$ of the proposed network consists of classical cross entropy loss $\mathcal{L}_{sem}$ and binary within-task weighted loss $\mathcal{L}_{dis}$ inspired by the work in [52]. Specifically, the overall weighted loss function $\mathcal{L}$ is derived as follows:

$$\mathcal{L} = \mathcal{L}_{sem} + \mathcal{L}_{dis} \tag{2}$$

where $\mathcal{L}_{dis}$ is expressed by a discriminative function [53] and written by

$$\mathcal{L}_{dis} = \omega_{rb} \cdot \mathcal{L}_{rb} + \omega_{rt} \cdot \mathcal{L}_{rt} \tag{3}$$

where $\mathcal{L}_{rb}$ and $\mathcal{L}_{rt}$ are defined with the classical cross entropy loss style [28] calculated in (4); ω_{rb} and ω_{rt} are weights of the network for the room boundary and type, calculated in (5).

$$\mathcal{L}_{task} = \frac{\left(\sum_{i=1}^{C} \hat{N}_i\right) - \hat{N}_i}{\sum_{j=1}^{C}\left(\left(\sum_{i=1}^{C} \hat{N}_i\right) - \hat{N}_j\right)} \cdot \left(\sum_{i=1}^{C} -y_i \log p_i\right),\ p_i \in [0,\ 1] \tag{4}$$

where $\hat{N}_i$ and C denote the number of ground-truths for the ith floor-plan primitive and the number of primitives in the corresponding task, respectively; and y_i and p_i are defined as the label and the prediction label for the ith floor-plan primitive.

$$\omega_{rb} = \frac{N_{rt}}{N_{rb} + N_{rt}},\ \omega_{rt} = \frac{N_{rb}}{N_{rb} + N_{rt}} \tag{5}$$

where N_{rb} and N_{rt} are the number of pixels for a boundary and room, respectively.

The final loss is a weighted summation after we train the two branches jointly. At the time of inference, the semantic labels of floor-plan primitives are obtained by applying an *argmax* operation.

3) Geometric Primitive Identification: As simple pixel-level semantic segmentation for a floor-plan raster map is far from satisfying, the proposed method further extracts the junctions from a rasterized image with information regarding walls and openings. Specifically, the junction detection pipeline is borrowed from an existing FCN [57]. The official implementation of the FCN was adopted in our junction detection experiments. Similar to [33], the Inception model of Google [54] is employed for encoding the input, while dividing the inputs into multiple $H_b \times W_b$ grids. Given a bin, the confidence score c_{conf} and junction coordinates (x, y) are predicted at individual output cells in the grid. During training, only junctions with $c_{conf} \geq 0.3$ are maintained. As shown in Fig. 5, based on the mature FCN architecture, junctions that are composed of wall corners and opening end-points are easily obtained.

The most representative primitives in indoor spaces are room regions, which are crucial for topological construction. Room regions are detected by using a standard instance segmentation technique [55]. Most methods detect objects through the interpolation of the predicted mask coefficients, which undoubtedly has a great impact on the instance segmentation task. Instance segmentation is essentially a pixel-level clustering problem. The mask boundary will be extremely rough when the interpolation multiple is too large. As demonstrated in [56], affinity is efficient for pixel-level instance segmentation. Therefore, the pixel affinity network was chosen for detecting room regions. First, the semantic and pixel affinity information of the floor-plan images is used as auxiliary supervision. Then, the generated information

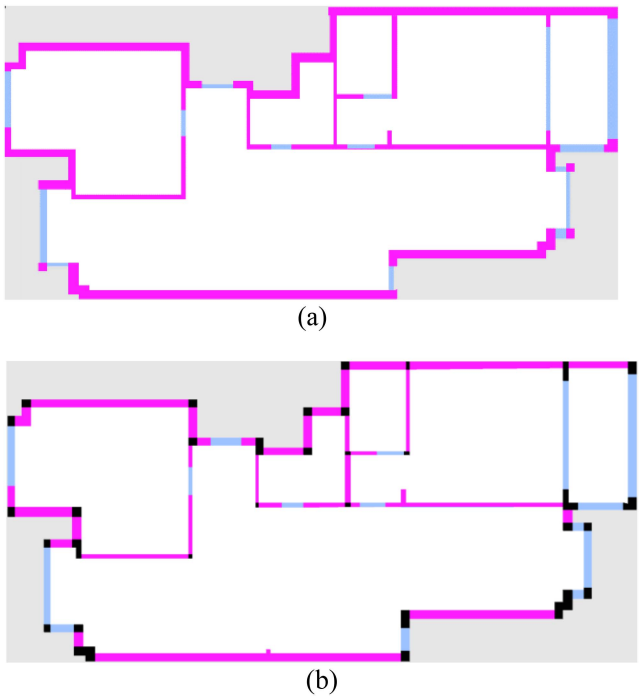

Fig. 5. Junctions recognition. (a) Primitive detection of walls (pink) and openings (light blue). (b) Recognition of junctions (black) using the FCN.

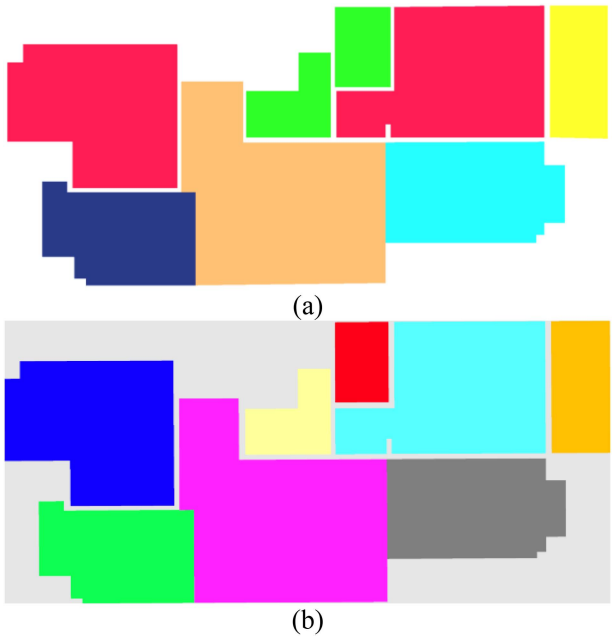

Fig. 6. Detecting regions. (a) Room semantics extraction. (b) Room region detection using semantic and pixel affinity information.

and the mask results are adopted a priori for instance mask optimization by initializing graph fusion. Simple primitive merging operations combine adjacent facets into one large region that represents a room in the floor-plan image, as shown in Fig. 6. Some skinny regions without any pixels with a room instance label are optionally merged to their adjacent rooms with the longest common boundary [37].

B. Indoor-Space Primitive Assembly With MIP

Geometric primitive assemblies of indoor spaces involve computing interior layouts. However, filtering out spurious primitives and guaranteeing floor-plan properties are challenging. Hence, a coarse-to-fine geometric primitive candidate assembly is adopted. Inspired by the work in [15], [33], and [38], the detected primitives and the associated relationship are fused into an MIP for information assemblies, as shown in Fig. 7.

According to [33], two types of pairwise primitive relationships are included: corner-to-edge (C2E) and region-to-region (R2R) relationships. For C2E relationships between a corner and an incident edge, the confidence score is acquired by the junction inference method of [57]. Determining R2R relationships is relatively complicated: given a set of room regions of floor-plan image, an improved Mask-RCNN [58] is utilized to identify sharing edges exclusively, which then serve as instance masks. C2E and R2R relationships are both employed by the subsequent objective function and relationship constraints.

After detecting geometric primitives and inferring primitive-wise relationships, the information is formulated as an objective function [59], which is solved by MIP for the primitive assembly of the topology and semantics representation.

1) Objective Function: The mixed integer quadratic problem is formulated by minimizing the extended objective function

$$\lambda_{data} \cdot E_{data} + \lambda_{relation} \cdot E_{relation} + \lambda_{prior} \cdot E_{prior} \quad (6)$$

where λ_{data}, $\lambda_{relation}$, and λ_{prior} are the weights that control the trade-off between the data term E_{data}, the relationship term $E_{relation}$, and the prior term E_{prior}, respectively, and where $\lambda_{data} = \lambda_{relation} = \lambda_{prior} = 1$.

The term E_{data} controls the consistency and complexity of the individual primitive (junction corner, boundary edge, and room region) representation by measuring the sum of geometric errors between each primitive to its corresponding structural element

$$\begin{aligned} E_{data} = \sum_{i,j} \alpha \cdot \Bigg\| \left(1 - \frac{\|c_{i,j} - c_{i,j-1}\|}{\|c_{i,j+1} - c_{i,j-1}\|}\right) \cdot c_{i,j-1} \\ + \frac{\|c_{i,j} - c_{i,j-1}\|}{\|c_{i,j+1} - c_{i,j-1}\|} \cdot c_{i,j+1} - c_{i,j} \Bigg\|^2 + \beta \cdot \left(1 - \frac{\tilde{e}_{i,j}}{e_{i,j}}\right) \cdot \frac{\hat{e}_{i,j}}{e_{i,j}} \\ + \gamma \cdot \left(0.5 \cdot \frac{r_{ij} - \tilde{r}_{ij}}{\hat{r}_{i,j} - \tilde{r}_{ij}} + 0.5 \cdot \left(1 - \frac{r_{ij}}{\hat{r}_{i,j}}\right)\right) \end{aligned} \quad (7)$$

where α, β, and γ are weights for each potential, and where $\alpha + \beta + \gamma = 1$; $c_{i,j}$, $e_{i,j}$, and $r_{i,j}$ represent the position of the *j*th junction, the length of the *j*th edge, and the area of the *j*th region in the *i*th floor-plan, respectively; $\tilde{e}_{i,j}$ is the length of the edge overlapping with the corresponding boundary element; $\hat{e}_{i,j}$ is the total length of all edges in the *i*th floor-plan; $\tilde{r}_{ij}$ is the area of the region of the inliers falling in the corresponding room element; and $\hat{r}_{i,j}$ is the total area of all regions in the *i*th floor-plan.

The term E_{rela} fuses the primitive-wise relationship information [33]

$$E_{relation} = \sum_{c,e,r} 10 \cdot (e_{conf} \cdot c'_{conf} \cdot c''_{conf} - 0.125) \cdot I_e(e)$$

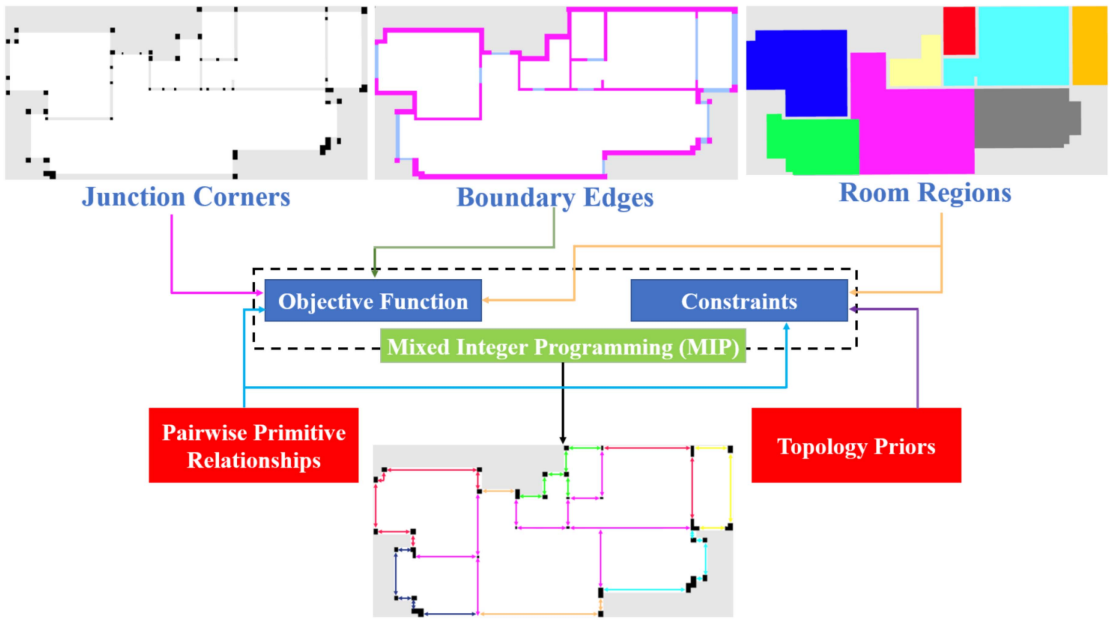

Fig. 7. Overview of the MIP for primitive assembly. The proposed pipeline inferred the relationships of detected primitives and fused the information into MIP to obtain a planar graph.

$$+ \left(\theta_{conf} \cdot c_{conf} - 0.25\right) \cdot I_c\left(\theta, c\right) + 0.1 \cdot \left(\phi_{conf} \cdot r_{conf} - 0.5\right) \cdot I_r\left(r\right) \tag{8}$$

where c_{conf}, e_{conf}, and r_{conf} are the corner (c), the edge (e), and the region (r) detection confidence scores, respectively; $I_c(\theta, c)$, $I_e(e)$, and $I_r(r)$ indicate indicator variables defined for the corner, edge, and region primitives, respectively; θ_{conf} and ϕ_{conf} are the C2E and R2R relationship confidence scores, respectively; and c' and c'' indicate the end-points of an edge.

The term E_{prior} encourages the size (w_i, d_i) and aspect ratio ε_i between the width (w_i) and the depth (d_i) of primitive i to become close to the best value, and is formulated as follows:

$$E_{prior} = \sum_i \left(\varepsilon_i - \varepsilon_i^*\right)^2 + \left(w_i - w_i^*\right)^2 + \left(d_i - d_i^*\right)^2 \tag{9}$$

where ε_i^* is the optimal ratio, and $(w_i^*,\ d_i^*)$ denotes the ground-truth size of primitive i.

2) Constraints: Relationship constraint: Relationship constraints are similar to the region constraints proposed in [60], but more powerful. In [33], a relationship constraint is a simple correlation of corners, edges, and regions, including inside constraints, mutual exclusion constraints, and position constraints. The concept of a relationship constraint is intuitive, but requires complex mathematical formulations. Inside constraints stipulate that all primitives must be inside the boundary of the given layout domain. That is, the region must be surrounded by edges; the expression in mathematical language is that the indicator variables $(I_e(e),\ I_r(r))$ of intersecting edges and regions must not be active at the same time $(\sum_{e \in \mathcal{E}_{e,r}} I_e(e) I_r(r) = 0$, where $\mathcal{E}_{e,r}$ denotes a set of edges that intersect a region r). Mutual exclusion constraints indicate that no overlapping occurs any pair of primitives and dummy primitives are not chosen simultaneously with the original one when they are spatially close; one of the edges that intersect with the last line segment must be the boundary edge $(\sum_{e \in \mathcal{E}_{e,o}} I_e(e) = 1$, where $\mathcal{E}_{e,o}$ denotes a set of collected edge candidates that intersect with an orthogonal line segment o). Position constraints specify the approximate position for each primitive and guarantee that a specified primitive covers the specified positions (i.e., the opening must be on the wall), which enforces $I_e(e)$ to be consistently active with its corresponding $I_d(\theta, c)$ $(\sum_{e \in \mathcal{E}_{e,\theta}} I_e(e) = I_d(\theta, c)$, where $\mathcal{E}_{e,\theta}$ represents the set of all candidate edges in a direction θ within m degrees in angular distance, and m is the value of the angular distance when the two edges cannot be in the same position at the same time).

Besides the above-mentioned basic constraints, high-level topology prior constraints that include size control and a region smoothness constraint are also included to optimize the initial primitive assembly for the correct topology and semantics between primitives.

Size control: To precisely control the size range and specify object sizes for room icon primitives, an aspect ratio constraint that requires that the extracted primitives are not too wide or too narrow is firstly provided. The challenge of setting the size control is that there is no prior information regarding which optimal ratio (r_i^*) it is. Similar to [38], an auxiliary binary variable σ_i for each primitive i is introduced to regulate the orientation (i.e., horizontal $(\sigma_i = 1)$ and vertical $(\sigma_i = 0)$) of the corresponding primitive automatically. Taking a room as an example, the aspect ratio constraint is written as follows:

$$\begin{cases} r\prime_i \cdot w_i \le d_i + (w_i^* + d_i^*) \cdot \sigma_i \\ r''_i \cdot w_i \ge d_i - (w_i^* + d_i^*) \cdot \sigma_i \\ r\prime_i \cdot d_i \le w_i + (w_i^* + d_i^*) \cdot (1 - \sigma_i) \\ r''_i \cdot d_i \ge w_i - (w_i^* + d_i^*) \cdot (1 - \sigma_i) \end{cases} \tag{10}$$

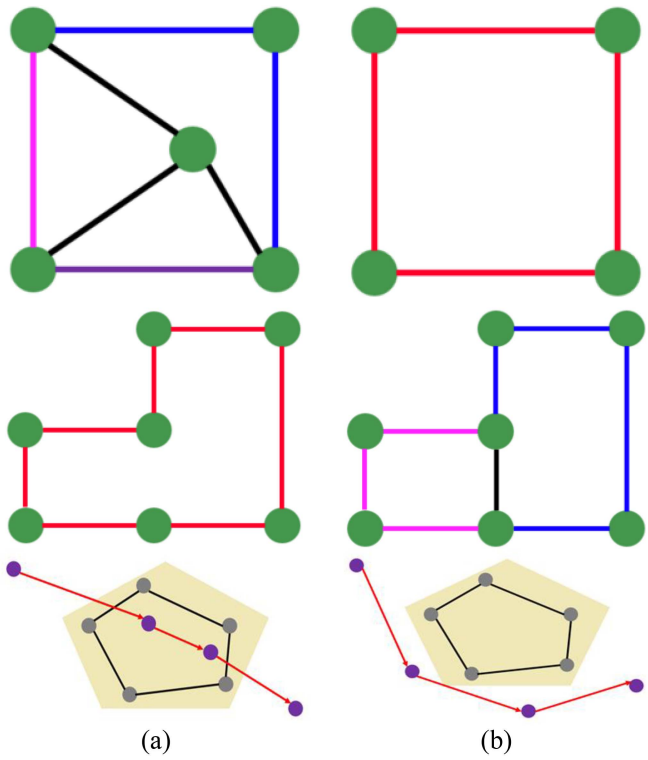

Fig. 8. Example of results before and after region smoothness. (a) Topology representation before region smoothness. (b) Topology representation after region smoothness. The first row is a merging operator, the second row is a splitting operator, and the third row is a topology correction.

where r'_i and r''_i are the minimum and maximum aspect ratios between w_i and d_i, respectively, of room icon primitive *i*, and the given domain (w_i, d_i) also possesses the minimum size (w'_i, d'_i) and maximum size $(w''_i, d''_i).(w_i^* + d_i^*)$, which is a predefined constant to ensure that the inequality is always established and satisfies the above-mentioned cases.

Region smoothness constraint: The shape of the output polygon is supposed to fulfill the manifoldness property [61], so the semantic information of each region must be constrained to be coherent with the probability map [62]. The coherence of regions is explored to refine constraints for extracting primitive candidates

$$\sum_{f \in F_x} -w_f \log P_{map}\left(l_f\right) \tag{11}$$

where w_f is the ratio of the area of region *f* to the area of the whole floor-plan image domain, and $P_{map}(l_f)$ is the mean of the probability map for class l_f over the pixels inside region *f* [63].

Fig. 8 shows an example of the region constraint for a correct representation. Two adjacent regions with different semantics are merged by removing the common edges, and one region is divided into multiple regions that have different semantics.

The extracted primitive representation is optimized in a hierarchical manner. The optimization stops when the indoor domain is not thoroughly covered by the assembled primitives, when the size error of primitives is larger than a specified threshold, when there is not enough space between a pair of primitives.

C. Graphic Reconstruction With a Polygonal Coordinate Descent Strategy

The output of the previous stage contains several topological conflicts between adjacent elements. To maintain the correct topology between primitives, the optimization problem is formulated as a polygonal loops reconstruction, with one polygonal loop for each primitive. Since walls are not consistently shared across rooms, a set of primitives are adopted to pare away the floor-plan graphical inference into the conversion of multiple polygonal curves with a loop topology. Thus, we can directly optimize the placement and the number of corners or end-points of a given wall.

The floor-plan is formulated as a polygonal loop set ($\{l_1, l_2, \ldots, l_n\}$, where *n* is the number of loops) optimization problem with an energy function [64] that balances geometric errors (data term), topological coherence (consistency term), and layout complexity (complexity term). The vectored structures are refined by minimizing the sum of the above-mentioned three objective terms

$$E\left(l_1, l_2, \ldots, l_n\right) = \sum_{i}^{n} E_{data}\left(L_i\right) + E_{consis}\left(l_1, l_2, \ldots, l_n\right) + \sum_{i}^{n} E_{complexity}\left(L_i\right), \tag{12}$$

The data term E_{data} states the sum of the geometric discrepancy with an input primitive over its corresponding basic pixels along each loop. The data term is a primitive-wise unary potential and includes two penalties, $E_{data}^{\mathcal{C}}$ and $E_{data}^{\mathcal{E}}$, written in (13). Specifically, the penalty $E_{data}^{\mathcal{C}}$ denotes one minus the pixel-wise corner likelihood [65], which encodes the place of a primitive corner at pixel *p*; the penalty $E_{data}^{\mathcal{E}}$ also defines one minus the pixel-wise edge likelihood, which encodes the place of an edge over pixel *p*.

$$E_{data}\left(L_i\right) = \sum_{p \in \mathbb{C}(L_i)} E_{data}^{\mathcal{C}}\left(p\right) + \sum_{p \in \mathbb{E}(L_i)} E_{data}^{\mathcal{E}}\left(p\right) \tag{13}$$

where L_i is a sequence of pixels at integer coordinates, and $\mathbb{C}(L_i)$ and $\mathbb{E}(L_i)$ are defined as the sum of corner pixels and edge pixels on L_i, respectively. The corner pixels and edge pixels of the given primitives are both obtained using Bresenham's line algorithm [8].

The consistency term E_{consis} is calculated by the consistency cost term introduced in [66]. The term denotes the number of pixels used by the primitive boundaries of all the polygonal loops together. If the neighboring primitives share boundaries (i.e., if two primitives are close to each other), E_{consis} goes down and otherwise goes up. In general, the consistency term imposes a penalty by moving two primitives to the same pixel such that it encourages the graphic loops to be topological consistent at the sharing boundaries. The consistency term is written as (14), and heavily penalizes inconsistency between shared corners and

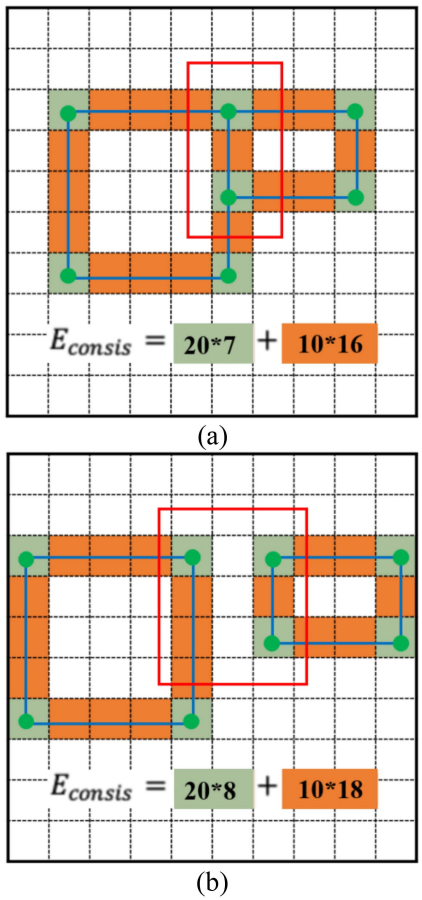

Fig. 9. Illustration of the consistency term E_{consis} counting the number of pixels occupied by primitive corners and edges. The red boxes refer to the difference between rooms adjacent or apart.

edge primitives.

$$E_{consis}(l_1, l_2, \ldots, l_n) = 20 \cdot \sum_p 1_{\mathcal{C}}(p, (l_1, l_2, \ldots, l_n)) + 10 \cdot \sum_p 1_{\mathcal{E}}(p, (l_1, l_2, \ldots, l_n)) \tag{14}$$

whereas the indicator variable for wall corners, $I_{\mathcal{C}}(p, (l_1, l_2, \ldots, l_n))$ is equal to 1 if the given pixel p belongs to a wall corner of at least one loop segment, and 0 otherwise. Similarly, $I_{\mathcal{E}}(p, (l_1, l_2, \ldots, l_n))$ is the indicator variable for edges. An example of calculating consistency terms is shown in Fig. 9.

The complexity term $E_{complexity}$ is calculated by counting the number of wall corners and the number of edges in the loops [66], where the lowest energy suggests that the room icon primitive contains fewer corners. The complexity term $E_{complexity}$ is written as follows:

$$E_{complexity} = \{N_{wc} + N_e\} \tag{15}$$

where N_{wc} refers to the number of wall corners and N_e represents the number of edges.

In the optimization procedure, each term is accompanied by an identical weight, which is learned by a grid search and remains unchanged throughout the processing. Finally, we optimize the reconstruction of multiple polygonal loops by minimizing the energy function $E(l_1, l_2, \ldots, l_n)$, calculated in (12), using a polygonal coordinate descent strategy [8]. The effect of the coordinate descent optimization strategy is shown in Fig. 10, where purple double arrows indicate the initial vector graphics representation after primitive assembly, fuchsia double arrows denote the random optimization output after the polygonal coordinate descent strategy, red double arrows represent the final vector graphics representation after multiple rounds of the coordinate descent optimization strategy, and red circles illustrate the errors repaired in the complex floor-plan structure.

IV. Experiments and Analysis

A. Implementation and Experimental Datasets

All of the experiments were conducted on a laptop equipped an NVIDIA GeForce RTX 3070 with 12 GB GPU main memory.

The proposed network was implemented using PyTorch[1] as the DNN library. To identify primitives, the resolution of the input floor-plan image was 512 × 512, and several hyperparameters (i.e., batch size, learning rate, momentum, and training time) were optimized during the training phase to determine the optimal combination by using a grid search approach. Specifically, the network was trained with 40 000 iterations with a batch size of 1 using the Adam optimizer to update parameters. Gurobi [38] was used in the indoor space primitive assembly to solve the MIP problem. The pixel-wise likelihoods for wall corners and edges in the graphical layer conversion were computed according to the method in [49]. Groups of five epochs were evaluated and the best one was chosen to find the optimal outputs.

To evaluate the performance of the proposed method, we took two public datasets (i.e., R2V dataset and R3D dataset) from [15] and [16], and collected additional floor-plan raster maps from Beike (www.ke.com) to prepare a new dataset with labels on various floor-plan elements. The R2V dataset comprised 870 ground-truth floor-plan images, randomly split into training and testing sets with roughly a 90/10 percentage split: 770 of them served as the training set and the remaining 100 images were used for testing. The R3D dataset was collected by crawling a rental website to collect 1259 photos. Roughly 10% of them had a more complex polygonal shape. The dataset consisted of more than 200 houses with approximately 1000 rooms and 7000 walls from urban and rural rental sites in London. In general, compared with the R2V and Beike datasets, the R3D dataset is more challenging, because it contains numerous irregular room shapes and a host of missing regions.

To more fully and efficiently exploit the data-hungry deep learning architecture, a custom dataset with a ground-truth for vector-graphics floor-plan conversions was built from Beike.[2] This custom dataset contained data from over 90 cities in China. To create the ground-truth, our team and other collaborators randomly sampled approximately 500 floor-plan raster maps and carefully annotated all data with geometric and semantic information, where the annotated information was then converted to

[1] [Online]. Available: https://pytorch.org/
[2] [Online]. Available: www.ke.com

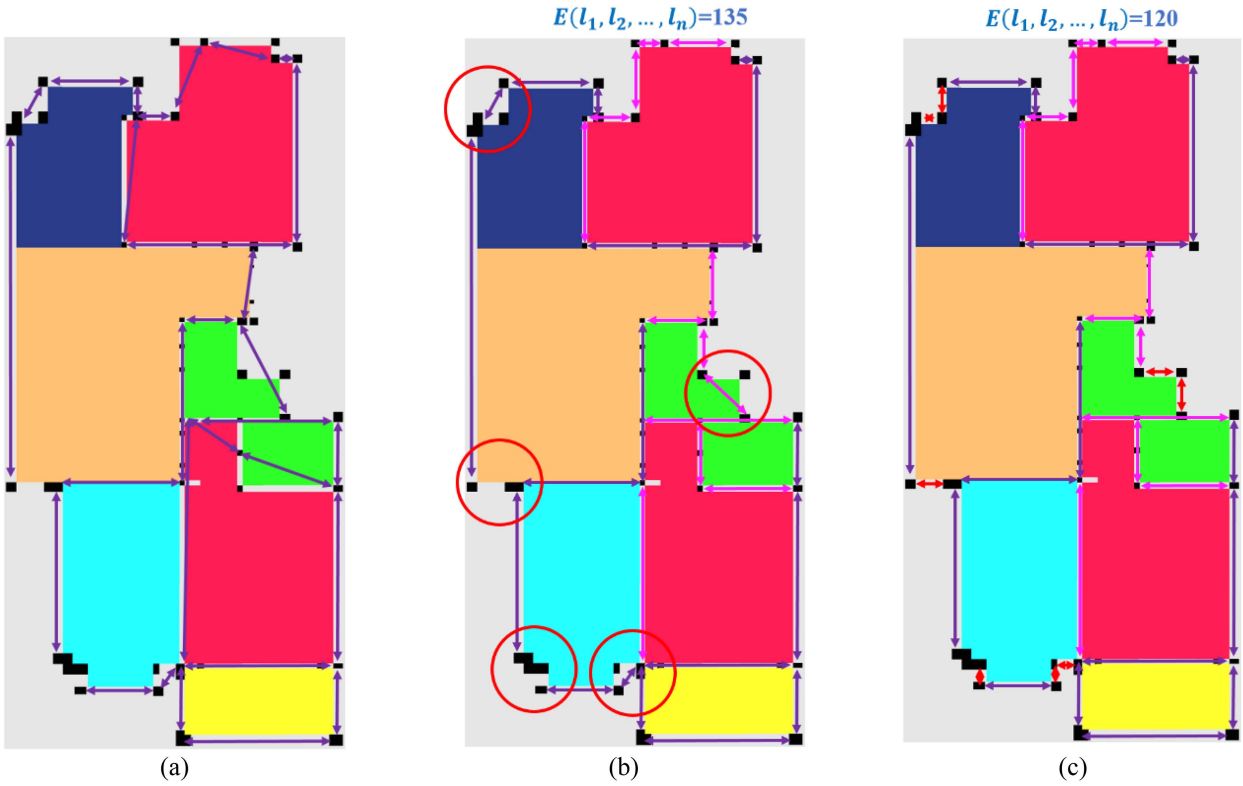

Fig. 10. Illustration of the vectorizing mistakes to fix the results using a polygonal coordinate descent strategy. (a) Initial vector graphics representation after primitive assembly. (b) Random optimization output after the polygonal coordinate descent strategy. (c) Final vector graphics representation after multiple rounds of the coordinate descent optimization strategy. Note that the value of the energy of each coordinate descent optimization is also shown.

our representation. In the implementation, we adopt the first 300 images as our training set, the next 50 images as a validation set, and the last 150 images as the test dataset.

B. Evaluation Criteria

The multiple aspects of the effectiveness of the proposed approach were quantitatively evaluated using four metrics (viz., *overall_accu* (OA), *average_accu* (AA), $F_\beta^{\max}$, and $F_\beta^{\min}$), which are widely used for the evaluation of interior building layout computations.

overall_accu and *average_accu* are the overall pixel accuracy and average of per-class pixel accuracy [67], calculated in (16) and (17), respectively.

$$overall\_accu = \frac{\sum_{i=1}^{k} N_i}{\sum_{i=1}^{k} \hat{N}_i} \tag{16}$$

$$average\_accu = \frac{N_i}{\hat{N}_i} \tag{17}$$

where k is the number of primitive types, and $\hat{N}_i$ and N_i denote the total number of ground-truths and the correct predictions for the ith floor-plan element, respectively.

$$F_\beta = \left(1+\beta^2\right)\left(Precision \cdot Recall\right)/\beta^2 \times \left(Precision + Recall\right), \tag{18}$$

$F_\beta^{\max}$ and F_β^{mean} are commonly used metrics [68], which are extended from the metric F_β [see (18)] [8], for quantitatively evaluating the binary maps produced from the proposed network output for walls pixels

$$F_\beta^{\max} = \frac{1}{M}\sum_{p=1}^{M} \tilde{F}_\beta^p \tag{19}$$

$$F_\beta^{\text{mean}} = \frac{1}{MT}\sum_{p=1}^{M}\sum_{t=0}^{T-1} F_\beta^p\left(\frac{t}{T-1}\right) \tag{20}$$

where t_{RCF} is a threshold to locate the walls from their results; M denotes the total number of testing floor-plan raster maps; $\tilde{F}_\beta^p$ represents the optimal F_β on the pth test input over T different t_{RCF} ranging in [0,1]; and $F_\beta^p(\frac{t}{T-1})$ is F_β on the pth test input using $t_{RCF} = \frac{t}{T-1}$. In the implementation, we set β^2=0.3 and $T = 256$ as suggested by a previous method [69].

The above-mentioned four metrics are used for evaluating the "semantics" modeling. To quantitatively evaluate vectorization results, the metric in [15] was adopted. If the minimum Euclidean distance between the current prediction result and the ground truth is less than the threshold, the current prediction is correct. For wall junctions, the threshold is τ_w, we set $\tau_w = \text{width}_{\text{wall}}/2$. For opening primitives, the distance between the prediction and the ground truth is the larger distance between the two pairs of corresponding endpoints [36], and the threshold is τ_o. We also set $\tau_o = \text{width}_{\text{wall}}/2$.

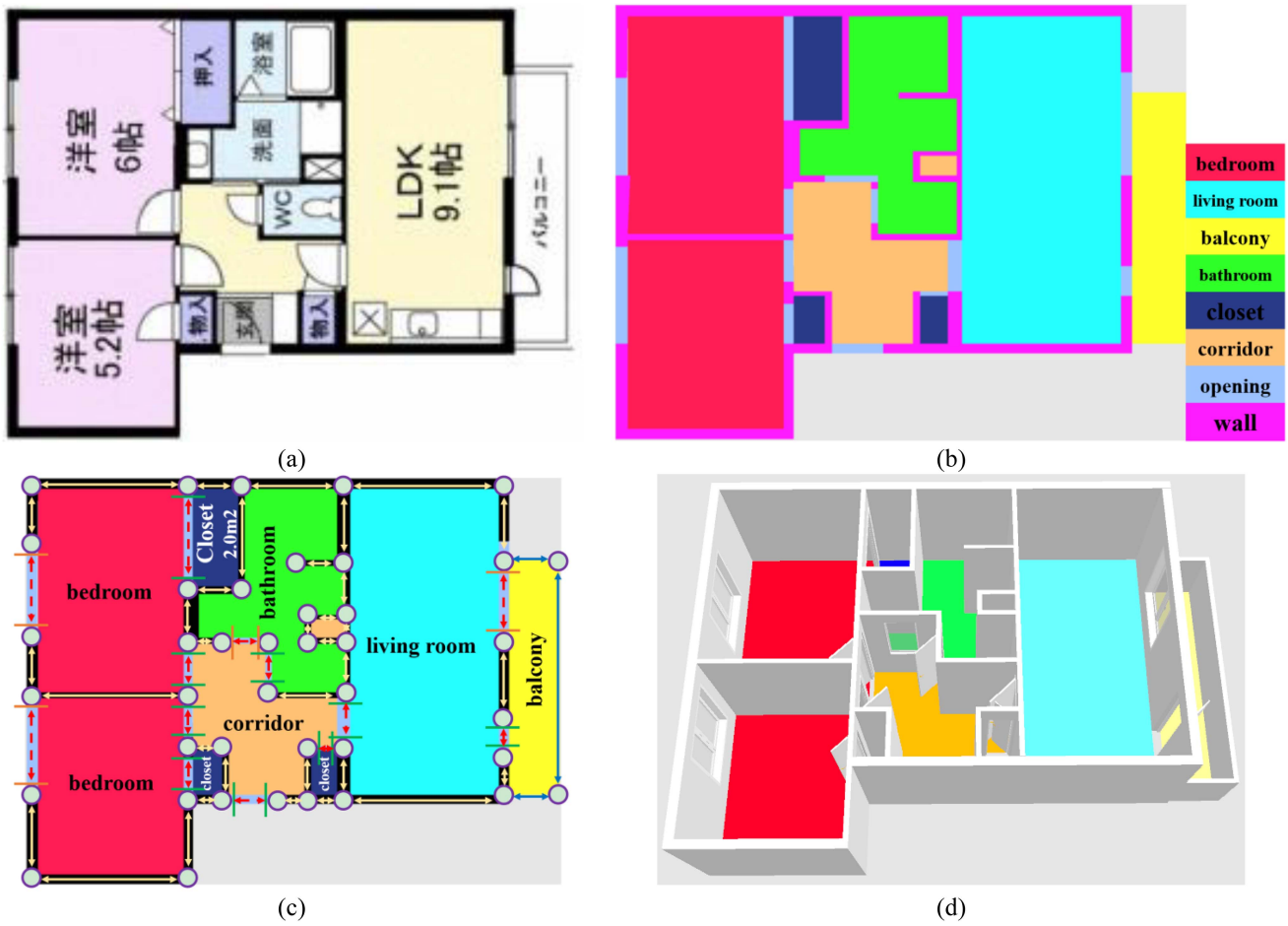

Fig. 11. Outcomes from R2V. (a) Floor-plan raster map. (b) Floor-plan element recognition. (c) Vector-graphics representation. (d) Popup 3D model.

TABLE I
QUANTITATIVE SEMANTIC EVALUATIONS BASED ON THREE DATASETS

Methods	R2V dataset				R3D dataset				Beike dataset			
	OA	AA	F_β^{max}	F_β^{mean}	OA	AA	F_β^{max}	F_β^{mean}	OA	AA	F_β^{max}	F_β^{mean}
[13]	0.87	**0.80**	/	/	0.81	0.72	/	/	/	/	/	/
[24]	0.85	0.74	0.81	0.79	**0.86**	0.77	0.79	0.79	/	/	/	/
The proposed method	**0.89**	0.76	**0.85**	**0.85**	**0.86**	**0.81**	**0.84**	**0.83**	**0.91**	**0.83**	**0.95**	**0.92**

C. Experimental Results and Comparisons

Figs. 11–13 show the outcomes of floor-plan element recognition, structured representation, and popup 3D model rendering for three selected floor-plan images from the R2V, R3D, and Beike datasets, respectively. Figs. 11–13(a) show the three selected floor-plan images, colored by the RGB values of each pixel. Figs. 11–13(b) are floor-plan element candidate recognition outcomes, dotted in different colors. Figs. 11–13(c) show the reconstructed vector representation with the area value of each primitive, where purple hollow circles, light yellow double arrow solid lines, and red double arrow dashed lines represent the junctions, walls, and opening primitives, respectively. The background of each room is dotted in different colors based on its inferred types. Figs. 11–13(d) show the popup 3D models, which are generated by extruding wall primitives to a certain height, and adding window and door textures at the location of opening primitives (when the faces of the given opening inside, the opening becomes a door).

The semantic representation of the proposed method was evaluated on three different datasets in terms of the four semantic evaluation metrics listed in Table I. The results indicate that the proposed method achieved the overall accuracy of 0.89, 0.86, and 0.91; the average accuracy of 0.76, 0.81, and 0.83, the $F_\beta^{\max}$ of 0.85, 0.84, and 0.95, and an F_β^{mean} of 0.85, 0.83, and 0.92 for the R2V dataset, R3D dataset, and Beike dataset respectively. As verified by the quantitative evaluation, the proposed method is capable of dealing with various building floor-plan raster maps (e.g., textured backgrounds that mix Chinese, English, or Japanese characters). The vectorization performance evaluation results are shown in Table II. Particularly, the proposed method

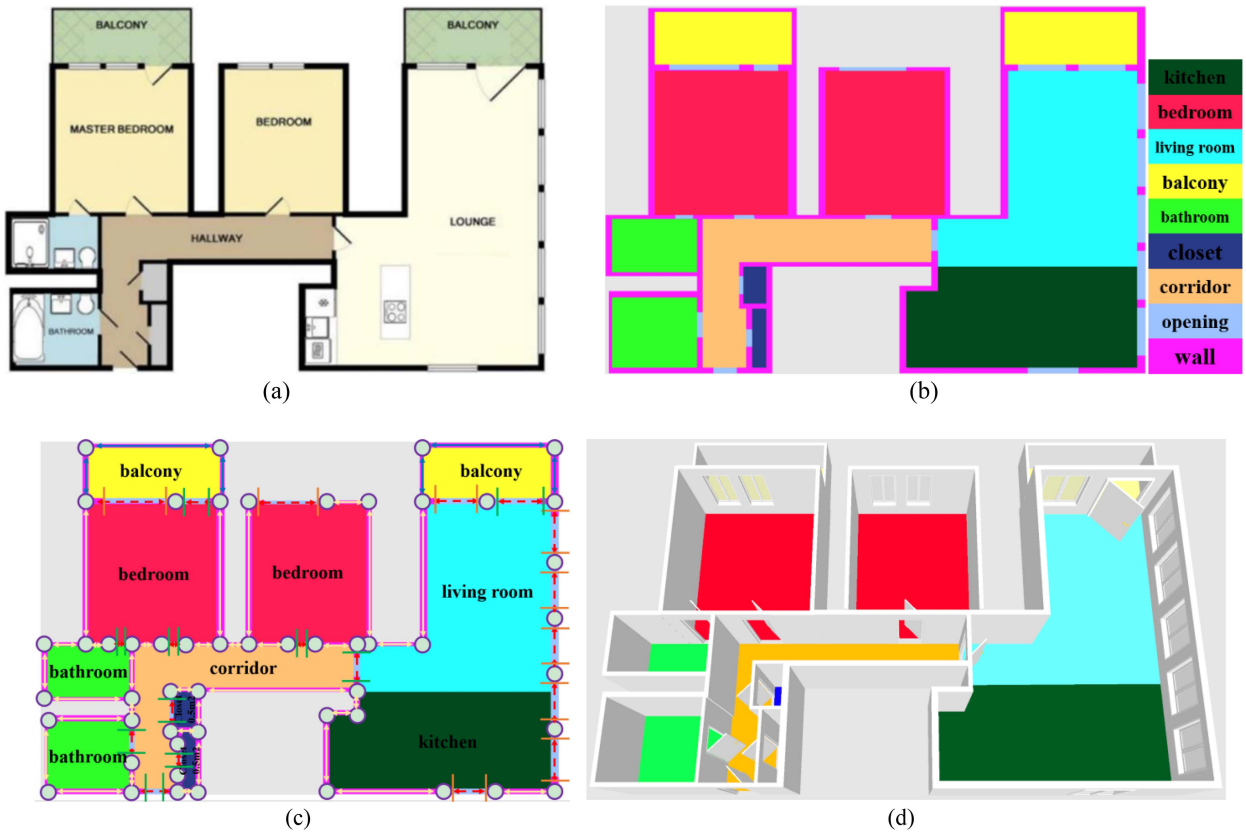

Fig. 12. Outcomes from R3D. (a) Floor-plan raster map. (b) Floor-plan element recognition. (c) Vector-graphics representation. (d) Popup 3D model.

TABLE II
QUANTITATIVE EVALUATIONS BASED ON THREE DATASETS

	R2V dataset				R3D dataset				Beike dataset			
	Wall Junction		Opening		Wall Junction		Opening		Wall Junction		Opening	
Methods	*Acc.*	*Rec.*	*Acc.*	*Rec.*	*Acc.*	*Rec.*	*Acc.*	*Rec.*	*Acc.*	*Rec.*	*Acc.*	*Rec.*
[15]	0.92	**0.92**	0.91	0.90	0.86	0.82	0.88	**0.91**	0.79	0.77	0.80	0.83
Our proposed method (w/o MIP and CD)	0.70	0.78	0.92	0.84	0.81	0.79	0.83	**0.91**	0.73	0.57	0.70	0.65
Our proposed method (w/o HC and CD)	0.89	0.88	0.91	0.87	0.82	0.83	0.88	0.86	0.79	0.78	0.75	0.73
Our proposed method (w/o CD)	0.92	0.90	**0.95**	0.92	0.89	**0.90**	0.90	0.88	0.84	0.79	0.86	0.85
Our proposed method complete version	**0.95**	0.91	**0.95**	**0.95**	**0.90**	0.89	**0.94**	**0.91**	**0.89**	**0.82**	**0.90**	**0.91**

HC and CD indicate high-level constraints (see Section III-B) and the coordinate descent strategy (see Section III-C), respectively. Acc. and Rec. refer to accuracy and recall, respectively.
Note: [28] cannot obtain vectorized results, so only [15] is compared.

achieves good performance in various complex indoor scenes, including various types of indoor layouts.

To further verify the impact of the individual procedure on interior layout representations, Table II also lists the performance of the proposed method with various procedures removed for ablation studies. To demonstrate and investigate the effectiveness of the proposed MIP approach, basic constraints or high-level constraints were disabled each time to record the performance. When full MIP was performed, the performance was consistently enhanced over the results without MIP. Specifically, the introduction of basic constraints filtered out many false primitives with only a small sacrifice in room icon primitives, improving the overall accuracy by a large margin. On the contrary, the performance for walls and openings was enhanced since more accurate layout structures were formed by the high-level constraints. As listed in Table II, the proposed approach obtained additional improvements in *Wall Junction* and *Opening* of three datasets with the proposed primitive-wise coordinate descent strategy for graph structure optimization.

Specifically, the proposed method revised a publicly available DNN. A spatial contextual mechanism was then introduced to enhance spatial semantics learning with a boundary-guided

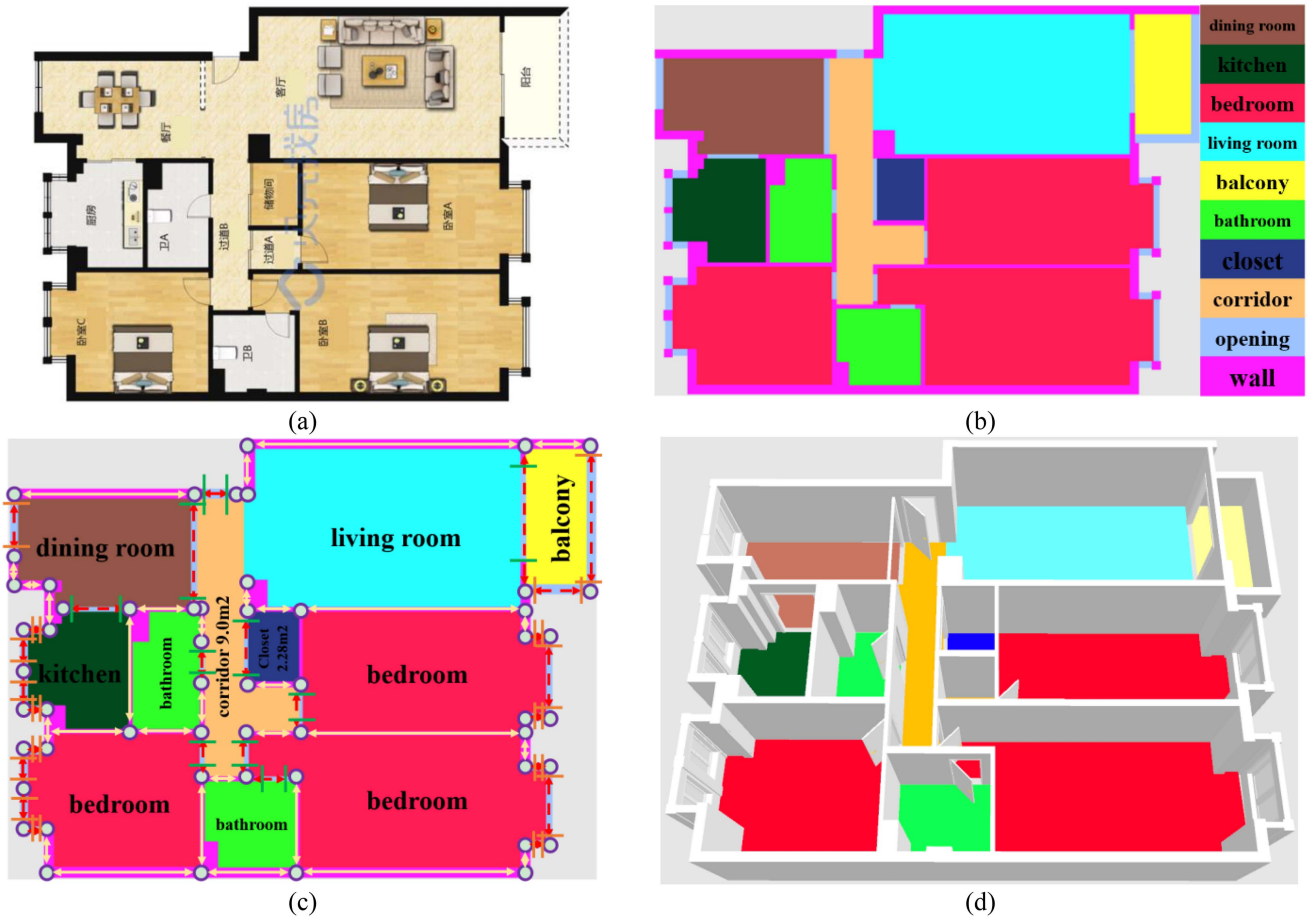

Fig. 13. Outcomes from Beike. (a) Floor-plan raster map. (b) Floor-plan element recognition. (c) Vector-graphics representation. (d) Popup 3D model.

attention module. The proposed method recognizes diverse floor-plan elements and further identifies a set of primitives and junctions with low-level semantics and geometries in most floor-plan datasets. The results [Figs. 11–13(b)] demonstrate that it is easy to differentiate inside and outside regions, even if there are some special room structures. The proposed method also correctly recognizes large wall elements as room primitives. An MIP is then formulated to join junctions to locate the primitives to generate a vector-graphics representation, while ensuring a correct topology and geometry. Further, it integrates graph structure optimization, improving the vectorized representation of interior building spaces. The final outputs [Figs. 11–13(c)] show that the proposed method obtains excellent results, not only in regular indoor scenes from floor-plan raster images, but also in terms of retrieving the primitives of complex structures. The reported metric values show that the proposed method performs well at computing interior building layouts, because we attach importance to the spatial relations among structural elements in the inference.

The methods in [15] and [28] have been implemented as baselines, so the proposed method was compared with the baselines in terms of the metrics mentioned in Section IV.B on the experimental datasets, as also listed in Tables I and II. For a fair comparison, we ran baselines, as well as the proposed method, on the two experimental datasets, and adjusted their own hyper-parameters to obtain the optimal results. Some metrics numbers missing for other methods are caused by the limitation of the compared methods instead of datasets. Fig. 14 illustrates the qualitative results of [15] and those of the proposed method on the R2V dataset. It can be seen from the red ovals that the proposed method identifies primitives without the Manhattan assumption, overcoming the limitation of [15]. It can be seen from Fig. 15 that both methods achieve high-quality elements from raster maps. Fig. 15 also shows that the proposed method achieves correct semantics and topology representations of indoor spaces.

The comparisons report that performance of the proposed method is better than those of [15] and [28]. The reason is that the proposed method takes advantages of [15] and [28] for high-quality primitive identification and semantics learning. The proposed method also deeply integrates graph structure optimization, improving vectorized representation for interior building layouts.

D. Performance

To show the performance of the proposed method in terms of computational cost, Table III provides the processing time

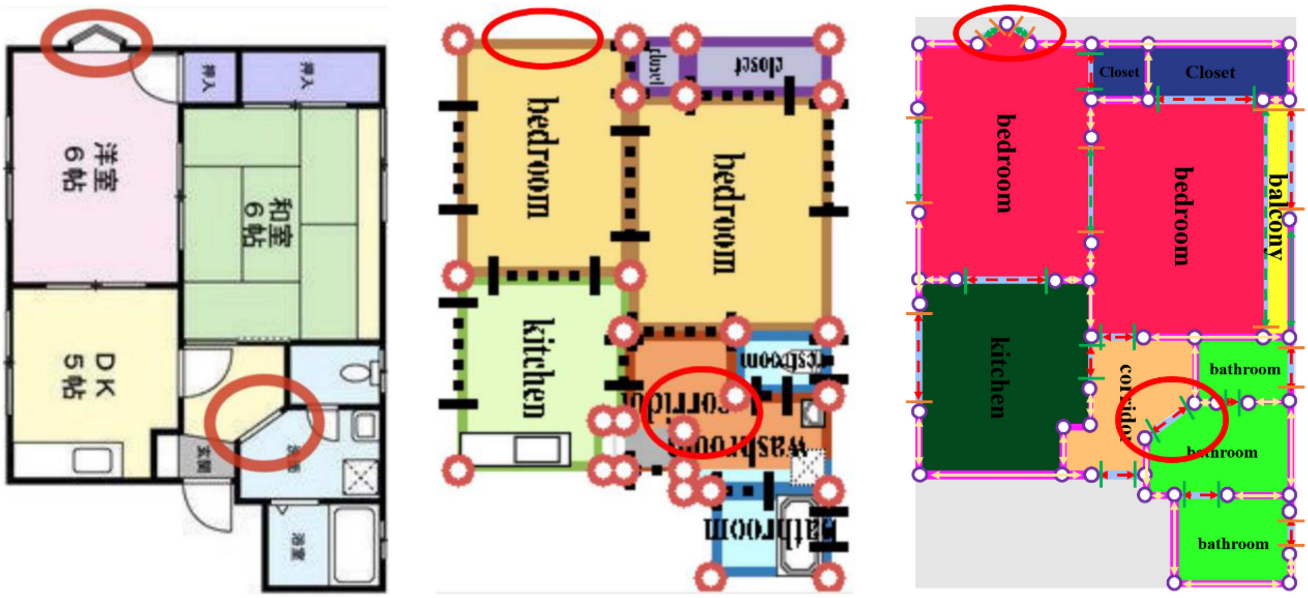

Fig. 14. Qualitative results of [15] and the proposed method on the R2V dataset. (a) Floor-plan raster map. (b) Vector-graphics representation of [15]. (c) Vector-graphics representation of the proposed method.

Fig. 15. Qualitative results of [28] and the proposed method on the R2V dataset. (a) Floor-plan raster map. (b) the results of [28]. (c) Vector-graphics representation with semantics and topology with the proposed method.

TABLE III
AVERAGE INFERENCE TIME OF EACH STEP FOR A FLOOR-PLAN IMAGE

	Primitives identification			Primitive assembly		Optimization	
	Sem. Seg	*J. B. Iden.*	*Inst. Seg.*	*Rel. Clas.*	*MIP*	*Topo. Opt.*	Total
Time (s)	1.83	5.69	3.67	0.96	178.35	134.78	325.28
Proportion (%)	0.56	1.75	1.13	0.30	54.82	41.44	100

Sem. Seg., J. B. Iden., Inst. Seg., and Rel. Clas. indicate pixel-level semantic segmentation, junction corners and boundary edges identification, instance segmentation for room region identification, and primitive-wise relationship classification, respectively.

of the different procedures during inference. Training usually takes approximately three days for the primitive detectors and relationship classifiers. The subsequent inference is much faster. At the time of inference, *MIP* and *Topo. Opt.* (i.e., the coordinate descent strategy for topology optimization) are the two most time-consuming steps, often requiring more than 95% of the total time. With several complex layouts, this can take up to 30 min (59 min in manual, 36 min in [15], respectively). However, the increased time enables us to acquire higher-quality representations. Indeed, the speed of the proposed method can be

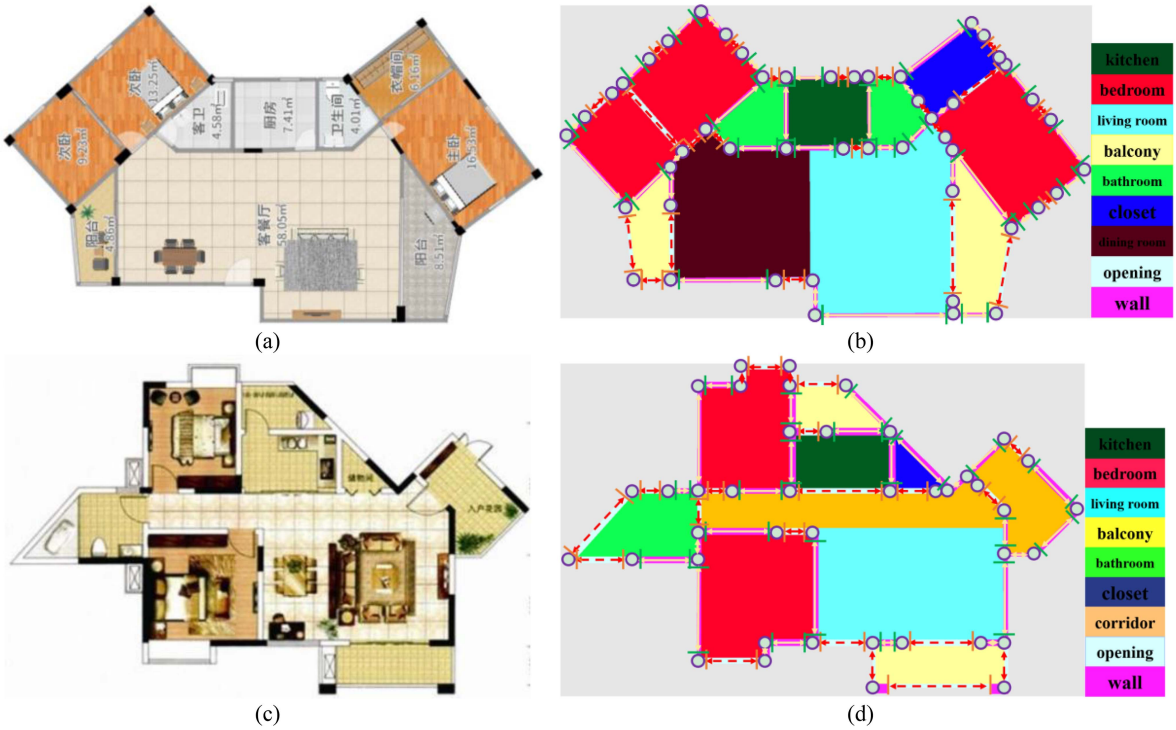

Fig. 16. Outcomes for boundaries of various shapes. (a) Floor-plan raster map. (b) Vector-graphics representation.

further reduced by special treatments (e.g., parallel process) or algorithm improvements, especially for the primitive assembly and topology optimization.

To evaluate its robustness, the proposed method was also tested on modern buildings that have walls that meet at 45 or even 30°. As illustrated in Fig. 16, the semantic and topology representation is correct, demonstrating the applicability of the proposed method to diverse floor-plans.

V. Conclusion

Automated semantic and topology representations of interiors of buildings from floor-plan raster maps are meaningful for indoor-space reconstruction and space analysis-related applications. We proposed a novel method of parsing and vectorizing indoor architecture from floor-plan raster maps. The proposed method adopts a learning-based hierarchical approach to identify a set of geometric primitives with semantics. The experimental results demonstrated that the proposed hierarchical approach is capable of effectively encoding features with significantly enhanced geometric primitive detection performance by revising existing networks. Then, the MIP is used to fuse primitives and their relationship information into vector graphics while enforcing high-level structural constraints, ultimately producing outputs that are close to the correct representation. Finally, these outputs are further enriched and refined with global energy optimization, with which we can obtain a vectorized floor-plan with the correct topology and semantics. Our comprehensive evaluation demonstrated the flexibility and robustness of the proposed method in three different datasets. The main limitations to the method are that it requires an extensive adjustment of the parameters and complex problem formulations with limited structural inference or constraints. These limitations will form the basis of our future work.

References

[1] A. A. Diakité and S. Zlatanova, "Spatial subdivision of complex indoor environments for 3D indoor navigation," *Int. J. Geographical Inf. Sci.*, vol. 32, no. 2, pp. 213–235, Feb. 2018.

[2] W. Zhang, W. Zhang, and J. Gu, "Edge-semantic learning strategy for layout estimation in indoor environment," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2730–2739, Jun. 2020.

[3] S. Yang, F. Wang, C. Peng, P. Wonka, M. Sun, and H. Chu, "Dula-Net: A dual-projection network for estimating room layouts from a single RGB panorama," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2019, pp. 3363–3372.

[4] J. Huang, Z. Kuang, F. Zhang, and T. Mu, "WallNet: Reconstructing general room layouts from RGB images," *Graph. Models*, vol. 111, Sep. 2020, Art. no. 101076.

[5] Y. Nie et al., "Shallow2deep: Indoor scene modeling by single image understanding," *Pattern Recognit.*, vol. 103, Jul. 2020, Art. no. 107271.

[6] C. Wang et al., "Semantic line framework-based indoor building modeling using backpacked laser scanning point cloud," *ISPRS J. Photogrammetry Remote Sens.*, vol. 143, pp. 150–166, Sep. 2018.

[7] S. Ochmann, R. Vock, and R. Klein, "Automatic reconstruction of fully volumetric 3D building models from oriented point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 151, pp. 251–262, May 2019.

[8] J. Chen, C. Liu, J. Wu, and Y. Furukawa, "Floor-SP: Inverse CAD for floorplans by sequential room-wise shortest path," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2661–2670.

[9] K. Khoshelham, H. Tran, D. Acharya, L. Díaz Vilarino, Z. Kang, and S. Dalyot, "Results of the ISPRS benchmark on indoor modelling," *ISPRS Open J. Photogrammetry Remote Sens.*, vol. 2, Dec. 2021, Art. no. 100008.

[10] P. Hübner, M. Weinmann, S. Wursthorn, and S. Hinz, "Automatic voxel-based 3D indoor reconstruction and room partitioning from triangle meshes," *ISPRS J. Photogrammetry Remote Sens.*, vol. 181, pp. 254–278, Nov. 2021.

[11] T. Jiang, Y. Wang, S. Liu, Y. Cong, L. Dai, and J. Sun, "Local and global structure for urban ALS point cloud semantic segmentation with ground-aware attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5702615.

[12] Y. Wang, T. Jiang, M. Yu, S. Tao, J. Sun, and S. Liu, "Semantic-based building extraction from LiDAR point clouds using contexts and optimization in complex environment," *Sensors*, vol. 20, no. 12, Jun. 2020, Art. no. 3386.

[13] T. Jiang, J. Sun, S. Liu, X. Zhang, Q. Wu, and Y. Wang, "Hierarchical semantic segmentation of urban scene point clouds via group proposal and graph attention network," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 105, Dec. 2021, Art. no. 102626.

[14] G. Laignel, N. Pozin, X. Geffrier, L. Delevaux, F. Brun, and B. Dolla, "Floor plan generation through a mixed constraint programming-genetic optimization approach," *Autom. Construction*, vol. 123, Mar. 2021, Art. no. 103491.

[15] C. Liu, J. Wu, P. Kohli, and Y. Furukawa, "Raster-to-vector: Revisiting floorplan transformation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2214–2222.

[16] C. Liu, A. Schwing, K. Kundu, R. Urtasun, and S. Fidler, "Rent3D: Floorplan priors for monocular layout estimation," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2015, pp. 3413–3421.

[17] P. Dosch, K. Tombre, C. Ah-Soon, and G. Masini, "A complete system for the analysis of architectural drawings," *Int. J. Document Anal. Recognit.*, vol. 3, pp. 102–116, Dec. 2000.

[18] T. Lu, H. Yang, R. Yang, and S. Cai, "Automatic analysis and integration of architectural drawings," *Int. J. Document Anal. Recognit.*, vol. 9, pp. 31–47, Jan. 2007.

[19] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, "Automatic room detection and room labeling from architectural floor plans," in *Proc. 10th IAPR Int. Workshop Document Anal. Syst.*, Mar. 2012, pp. 339–343.

[20] N. Nauata, S. Hosseini, K. Chang, H. Chu, C. Cheng, and Y. Furukawa, "House-GAN++: Generative adversarial layout refinement networks," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2021, pp. 13632–13641.

[21] L. Gimenez, S. Robert, F. Suard, and K. Zreik, "Automatic reconstruction of 3D building models from scanned 2D floor plans," *Autom. Construction*, vol. 63, pp. 48–56, Mar. 2016.

[22] L. P. de las Heras, S. Ahmed, M. Liwicki, E. Valveny, and G. Sanchez, "Statistical segmentation and structural recognition for floor plan interpretation," *Int. J. Document Anal. Recognit.*, vol. 17, pp. 221–237, Feb. 2014.

[23] L. Gimenez, J. Hippolyte, S. Robert, F. Suard, and K. Zreik, "Review: Reconstruction of 3D building information models from 2D scanned plans," *J. Building Eng.*, vol. 2, pp. 24–35, Feb. 2015.

[24] L. Yang and M. Worboys, "Generation of navigation graphs for indoor space," *Int. J. Geographical Inf. Sci.*, vol. 29, no. 10, pp. 1737–1756, Oct. 2015.

[25] J. Song and K. Yu, "Framework for indoor elements classification via inductive learning on floor plan graphs," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 2, Feb. 2021, Art. no. 97.

[26] S. Dodge, J. Xu, and B. Stenger, "Parsing floor plan images," in *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl.*, 2017, pp. 358–361.

[27] T. Yamasaki, J. Zhang, and Y. Takada, "Apartment structure estimation using fully convolutional networks and graph model," in *Proc. ACM Workshop Multimedia for Real Estate Tech*, 2018, pp. 1–6.

[28] Z. Zeng, X. Li, Y. Yu, and C. Fu, "Deep floor plan recognition using a multi-task network with room-boundary-guided attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9095–9103.

[29] G. Renton, P. Héroux, B. Gaüzère, and S. Adam, "Graph neural network for symbol detection on document images," in *Proc. Int. Conf. Document Anal. Recognit. Workshops*, 2019, pp. 62–67.

[30] J. Shang, X. Tang, F. Yu, and F. Liu, "A semantics-based approach of space subdivision for indoor fine-grained navigation," *J. Comput. Inf. Syst.*, vol. 11, no. 9, pp. 3419–3430, Sep. 2015.

[31] M. Previtali, L. Díaz-Vilariño, and M. Scaioni, "Indoor building reconstruction from occluded point clouds using graph-cut and ray-tracing," *Appl. Sci.*, vol. 8, Sep. 2018, Art. no. 1529.

[32] J. Han, M. Rong, H. Jiang, H. Liu, and S. Shen, "Vectorized indoor surface reconstruction from 3D point cloud with multistep 2D optimization," *ISPRS J. Photogrammetry Remote Sens.*, vol. 177, pp. 57–74, Jul. 2021.

[33] N. Nauata and Y. Furukawa, "Vectorizing world buildings: Planar graph reconstruction by primitive detection and relationship inference," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 711–726.

[34] F. Zhang, N. Nauata, and Y. Furukawa, "Conv-MPN: Convolutional message passing neural network for structured outdoor architecture reconstruction," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2020, pp. 2795–2804.

[35] C. Liu, J. Wu, and Y. Furukawa, "FloorNet: A unified framework for floorplan reconstruction from 3D scans," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 203–219.

[36] X. Lv, S. Zhao, X. Yu, and B. Zhao, "Residential floor plan recognition and reconstruction," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2021, pp. 16717–16726.

[37] H. Fang, C. Pan, and H. Huang, "Structure-aware indoor scene reconstruction via two levels of abstraction," *ISPRS J. Photogrammetry Remote Sens.*, vol. 178, pp. 155–170, Aug. 2021.

[38] W. Wu, L. Fan, L. Liu, and P. Wonka, "MIQP-based layout design for building interiors," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 511–521, May 2018.

[39] W. Wu, X. Fu, R. Tang, Y. Wang, Y. Qi, and L. Liu, "Data-driven interior plan generation for residential buildings," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 234:1–234:12, Nov. 2019.

[40] Y. Wu, J. Shang, P. Chen, S. Zlatanova, X. Hu, and Z. Zhou, "Indoor mapping and modeling by parsing floor plan images," *Int. J. Geographical Inf. Sci.*, vol. 35, no. 6, pp. 1205–1231, Jun. 2021.

[41] H. Jang, K. Yu, and J. Yang, "Indoor reconstruction from floorplan images with a deep learning approach," *Int. J. Geographical Inf. Sci.*, vol. 9, no. 2, Jan. 2021, Art. no. 65.

[42] R. Hu, Z. Huang, Y. Tang, O. V. Kaick, H. Zhang, and H. Huang, "Graph2Plan: Learning floorplan generation from layout graphs," *ACM Trans. Graph.*, vol. 39, no. 4, pp. 118:1–118:4, Jul. 2020.

[43] N. Nauata, K. Chang, C. Cheng, G. Mori, and Y. Furukawa, "House-GAN: Relational generative adversarial networks for graph-constrained house layout generation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 162–177.

[44] M. Vidanapathirana, Q. Wu, Y. Furukawa, A. X. Chang, and M. Savva, "Plan2Scene: Converting floorplans to 3D scenes," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2021, pp. 10733–10742.

[45] J. Yang, Z. Kang, L. Zeng, P. Akwensi, and M. Sester, "Semantics-guided reconstruction of indoor navigation elements from 3D colorized points," *ISPRS J. Photogrammetry Remote Sens.*, vol. 173, pp. 238–261, Mar. 2021.

[46] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style convnets great again," in *Proc. Comput. Vis. Pattern Recognit.*, 2021, pp. 13733–13742.

[47] C. Wen, X. Sun, J. Li, C. Wang, Y. Guo, and A. Habib, "A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 147, pp. 178–192, Apr. 2019.

[48] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 717–732.

[49] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 636–644.

[50] Y. Zhang, Y. He, S. Zhu, and X. Di, "The direction-aware, learnable, additive kernels and the adversarial network for deep floor plan recognition," 2020, *arXiv: 2001.11194*.

[51] P. Zhong, D. Wang, and C. Miao, "An affect-rich neural conversational model with biased attention and weighted cross-entropy loss," in *Proc. AAAI*, 2019, pp. 7492–7500.

[52] L. Zhao and W. Tao, "JSNet: Joint instance and semantic segmentation of 3D point clouds," in *Proc. AAAI*, 2020, pp. 12951–12958.

[53] Y. Wang, T. Jiang, J. Liu, X. Li, and C. Liang, "Hierarchical instance recognition of individual roadside trees in environmentally complex urban areas from UAV laser scanning point clouds," *ISPRS Int. J. Geoinf.*, vol. 9, no. 10, Oct. 2020, Art. no. 595.

[54] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-RNN++," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2018, pp. 859–868.

[55] M. Li, F. Lafarge, and R. Marlet, "Approximating shapes in images with low-complexity polygons," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2020, pp. 8630–8638.

[56] Y. Liu et al., "Affinity derivation and graph merge for instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 708–724.

[57] K. Huang, Y. Wang, Z. Zhou, T. Ding, S. Gao, and Y. Ma, "Learning to parse wireframes in images of man-made environments," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 626–635.

[58] A. Vuola, S. Akram, and J. Kannala, "Mask-RCNN and U-Net ensembled for nuclei segmentation," in *Proc. IEEE 16th Int. Symp. Biomed. Imag.*, 2019, pp. 208–212.

[59] F. Zhang, S. Xu, N. Nauata, and Y. Furukawa, "Structured outdoor architecture reconstruction by exploration and classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 12407–12415.

[60] Y. Qian and Y. Furukawa, "Learning pairwise inter-plane relations for piecewise planar reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 330–345.
[61] Y. Wang et al., "Interactive structure-aware blending of diverse edge bundling visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 687–696, Jan. 2020.
[62] M. Li, L. Nan, N. Smith, and P. Wonka, "Reconstructing building mass models from UAV images," *Comput. Graph.*, vol. 54, pp. 84–93, Feb. 2016.
[63] M. L. Li, F. Rottensteiner, and C. Heipke, "Modelling of buildings from aerial LiDAR point clouds using TINs and label maps," *ISPRS J. Photogrammetry Remote Sens.*, vol. 154, pp. 127–138, Aug. 2019.
[64] Z. Dong, B. Yang, P. Hu, and S. Scherer, "An efficient global energy optimization approach for robust 3D plane segmentation of point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 137, pp. 112–133, Mar. 2018.
[65] K. Yi, H. Jeong, B. Lee, and J. Choi, "Visual tracking in complex scenes through pixel-wise tri-modeling," *Mach. Vis. Appl.*, vol. 26, pp. 205–217, Jan. 2015.
[66] C. Lin, C. Li, and W. Wang, "Floorplan-jigsaw: Jointly estimating scene layout and aligning partial scans," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5673–5682.
[67] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[68] Y. Liu et al., "Richer convolutional features for edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1939–1946, Aug. 2019.
[69] Q. Hou, M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, Apr. 2019.

Bisheng Yang received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2002.

He is currently a Professor of Geomatics Engineering and the Vice-Director of the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. His research expertises include laser scanning and photogrammetry, point cloud processing, and GIS applications.

Dr. Yang received numerous national and international academic awards including the Carl Pulfrich Award (2019).

Tengping Jiang received the M.S. degree in cartography and geography information system from Nanjing Normal University, Nanjing, China, in 2019. He is currently working toward the Ph.D. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China.

His research interests include model reconstruction and related applications.

Weitong Wu received the B.E. degree in geomatics engineering from Central South University, Changsha, China, in 2017. He is currently working toward the Ph.D. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China.

His research interests include semantic and structure-aware multisensor fusion SLAM (simultaneous localization and mapping).

Yuzhou Zhou received the M.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2022. He is currently working toward the Ph.D. degree with the Department of Computer Science, University of Oxford, Oxford, U.K.

His research interests include LiDAR point cloud understanding and related applications.

Lei Dai received the M.S. degree in geomatics engineering in 2022 from Wuhan University, Wuhan, China, where he is currently working toward the Ph.D. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing.

His research interests include point cloud intelligent processing and related applications.